

Color Image Segmentation Using a Spatial K-Means Clustering Algorithm

Dana Elena Ilea and Paul F. Whelan

Vision Systems Group
School of Electronic Engineering
Dublin City University
Dublin 9, Ireland
danailea@eeng.dcu.ie

Abstract

This paper details the implementation of a new adaptive technique for color-texture segmentation that is a generalization of the standard K-Means algorithm. The standard K-Means algorithm produces accurate segmentation results only when applied to images defined by homogenous regions with respect to texture and color since no local constraints are applied to impose spatial continuity. In addition, the initialization of the K-Means algorithm is problematic and usually the initial cluster centers are randomly picked. In this paper we detail the implementation of a novel technique to select the dominant colors from the input image using the information from the color histograms. The main contribution of this work is the generalization of the K-Means algorithm that includes the primary features that describe the color smoothness and texture complexity in the process of pixel assignment. The resulting color segmentation scheme has been applied to a large number of natural images and the experimental data indicates the robustness of the new developed segmentation algorithm.

Keywords: K-Means clustering, Color extraction, diffusion based filtering, texture, color spaces.

1. Introduction

Image segmentation is one of the most important precursors for image processing-based applications and has a crucial impact on the overall performance of the developed systems. Robust segmentation has been the subject of research for many years, but the published work indicates that most of the developed image segmentation algorithms have been designed in conjunction with particular applications. The aim of the segmentation process consists of dividing the input image into several disjoint regions with similar characteristics such as color and texture. Robust image segmentation is a difficult task since often the scene objects are defined by image regions with non-homogenous texture and color characteristics and in order to divide the input image into semantically meaningful regions many developed algorithms either use *a priori* knowledge in regard to the scene objects or employ the parameter estimation for local texture [1]. The development of texture alone approaches proved to be limited and the use of color information in the development of joint color-texture models has led to the development of more robust and generic segmentation algorithms [2,3,4]. The area of color image analysis is one of the most active topics of research and a large number of color-driven segmentation techniques have been proposed. Most representative color segmentation techniques include histogram-based segmentation, probabilistic space partitioning and clustering [5], region growing, Markov random field and simulated annealing [6]. All these techniques have the aim to reduce the number of color components from the input image into a reduced number of components in the color segmented image that are strongly related to the image objects.

In this paper we have developed a new technique that is a generalization of the standard K-Means clustering technique. The K-Means clustering technique is a well-known approach that has been applied to solve low-level image segmentation tasks. This clustering algorithm is convergent and its aim is to optimize the partitioning decisions based on a user-defined initial set of clusters

that is updated after each iteration. This procedure is computationally efficient and can be applied to multidimensional data but in general the results are meaningful only if homogenous non-textured color regions define the image data. The applications of the clustering algorithms to the segmentation of complex color-textured images is restricted by two problems. The first problem is generated by the starting condition (the initialization of the initial cluster centers), while the second is generated by the fact that no spatial (regional) cohesion is applied during the space partitioning process. In this paper we developed a new space-partitioning scheme that addresses both these limitations.

The selection of initial cluster centers is very important since this prevents the clustering algorithm to converge to local minima, hence producing erroneous decisions. The most common initialization procedure selects the initial cluster centers randomly from input data. This procedure is far from optimal because does not eliminate the problem of converging to local minima and in addition the segmentation results will be different any time the algorithm is applied. To circumvent this problem some authors applied the clustering algorithms in a nested sequence but the experimental data indicated that this solution is not any better than the random initialization procedure. In this paper we propose a different approach to select the cluster centers by extracting the dominant colors from the color histograms. The developed procedure is generic and proved to be very efficient when applied to a large number of images. There are other initialization schemes proposed in the literature and for more details the reader can refer to [7].

The second limitation associated with the K-Means (and in general clustering algorithms) is generated by the fact that during the space partitioning process the algorithm does not take into consideration the local connections between the data points (color components of each pixel) and its neighbors. This fact will restrict the application of clustering algorithms to complex color-textured images since the segmented output will be over-segmented. To address this issue we have generalized the K-Means algorithm to evaluate along with the pixel color information two more distributions that sample the local color smoothness and the local texture complexity. In this regard to sample the local color smoothness, the image is filtered with an adaptive diffusion scheme while the local texture complexity is sampled by filtering the input image with a gradient operator. Thus, during the space partitioning process, the developed algorithm attempts to optimize the fitting of the diffusion-gradient distributions in a local neighborhood around the pixels under analysis with the diffusion (color)-gradient distributions for each cluster. This process is iteratively applied until convergence is reached. We have applied the developed spatial clustering algorithm on a large selection of images with different level of texture complexity and on test data that has been artificially corrupted with noise.

2. K-Means Algorithm

In general, spatial partitioning methods are implemented using iterative frameworks that either attempt to minimize the variation within the clusters or attempt to identify the optimal partitions based on a set of Gaussian Mixture Models. In this paper we focus on the implementation of the K-Means algorithm, although the methodology detailed in this paper can be applied to other clustering schemes such as fuzzy clustering [8] or competitive agglomerative clustering [9].

The K-Means is a nonhierarchical clustering technique that follows a simple procedure to classify a given data set through a certain number of K clusters that are known *a priori*. The K-Means algorithm updates the space partition of the input data iteratively, where the elements of the data are exchanged between clusters based on a predefined metric (typically the Euclidian distance between the cluster centers and the vector under analysis) in order to satisfy the criteria of minimizing the variation within each cluster and maximizing the variation between the resulting K clusters. The algorithm is iterated until no elements are exchanged between clusters. This clustering algorithm, in its standard formulation consists mainly of four steps that are briefly described below:

Steps of the classical K-Means clustering algorithm:

1. Initialization – generate the starting condition by defining the number of clusters and randomly select the initial cluster centers.
2. Generate a new partition by assigning each data point to the nearest cluster center.
3. Recalculate the centers for clusters receiving new data points and for clusters losing data points.
4. Repeat the steps 2 and 3 until a distance convergence criterion is met.

As mentioned before, the aim of the K-Means is the minimization of an objective function that samples the closeness between the data points and the cluster centers, and is calculated as follows:

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2 \quad (1)$$

where $\|x_i^{(j)} - c_j\|^2$ is the distance (usually the Euclidian metric) between the data point $x_i^{(j)}$ and the cluster center c_j . As it can be easily observed in equation 1, the assignment of the data points may not be unique (a data point can be equally distanced from two or more cluster centers) a case when the K-Means algorithm doesn't find the optimal solution corresponding to the global objective function J . In addition, it is sensitive to the initialisation process that selects the initial cluster centers (usually randomly picked from input data). If the initial cluster centers are initialised on outliers, the algorithm will converge to local minima and this is one of the major drawbacks of this space partitioning technique. More importantly this algorithm does not produce meaningful results when applied to noisy data or to tasks such as the segmentation of complex textured images or images affected by uneven illumination. Since the pixel assignment is performed only by evaluating the color information in a certain color space, the connection between the data point under evaluation and its neighbours is not taken into account, a fact that will lead to a partition of the input data into regions that are not related to the scene objects. In the remainder of this paper we will detail a histogram based procedure used to select the dominant colors from input data and the development of a new data assignment strategy that evaluates not only the pixel's color components but also the local color-texture complexity, which allows us to obtain homogenous clusters.

3. Automatic Seed Generation

Since the random selection of the initial cluster centers from image data is not an appropriate solution, we have developed a new scheme to perform the initialization for the K-Means algorithm with the dominant color components that are extracted from the color histograms of the input image. In this regard, we have constructed the histogram for each color channel and partitioned them linearly into R sections (where R is a fixed value, $R > K$ and n_k is the number of pixels contained in the bin k) and for each section of the histograms is determined the bin that has the highest number of elements:

$$P_j = \arg \max_{j \in [1, R]} (n_k) \quad (2)$$

We continue with ranking the peaks obtained from color histograms in agreement with the number of elements that are sorted in descending order. Finally, we form the color seeds (dominant colors) starting with the histogram peak that has the highest number of elements. This process can be summarized by the following pseudo-code sequence:

1. Construct the histograms for each color channel
2. Partition each histogram into R sections
3. Compute the peaks in each section and rank the peaks, p_1, p_2, \dots, p_R where p_1 has the highest number of elements
4. Start to form the color seeds for highest peak p_i
 - if($p_i \rightarrow$ red) mark the pixels in the red channel and calculate the g_{mean} and b_{mean} for marked pixels from green and blue channels
 - if($p_i \rightarrow$ green) mark the pixels in the green channel and calculate the r_{mean} and b_{mean} for marked pixels from red and blue channels
 - if($p_i \rightarrow$ blue) mark the pixels in the blue channel and calculate the r_{mean} and g_{mean} for marked pixels from red and green channels
5. Form the color seed and eliminate p_i from the list
6. Repeat the steps 4 and 5 until the desired number of color seeds has been reached.

4. Diffusion-based Filtering

As it was mentioned in Section 3, one of our aims is to sample the local color smoothness. This can be achieved by filtering the data with a smoothing operator that eliminates the weak textures. The standard linear smoothing filtering schemes based on Gaussian weighted spatial operators or non-linear filters such as the median, reduce the level of noise but this advantage is obtained at the expense of poor feature preservation (i.e. suppression of narrow details in the image). To circumvent this problem we have developed an adaptive diffusion based filtering scheme that was originally developed by Perona and Malik with the purpose of implementing an optimal, feature preserving smoothing strategy [10]. In their paper, smoothing is formulated as a diffusive process and is performed within the image regions and suppressed at the regions boundaries. This non-linear smoothing procedure can be defined in terms of the derivative of the flux function that is illustrated in equation 3.

$$u_t = \text{div}(D(|\nabla u|)\nabla u) \quad (3)$$

where u is the input data, D represents the diffusion function and t indicates the iteration step. The smoothing strategy described in equation 3 can be implemented using an iterative formulation as follows:

$$I_{x,y}^{t+1} = I_{x,y}^t + \lambda \sum_{j=1}^4 [D(\nabla_j I) \nabla_j I] \quad (4)$$

$$D(\nabla I) = e^{-\left(\frac{|\nabla I|}{d}\right)^2} \in (0,1] \quad (5)$$

where $\nabla_j I$ is the gradient operator defined in a 4-connected neighborhood, λ is the contrast operator that is set in the range $0 < \lambda < 0.16$ and d is the diffusion parameter that controls the smoothing level. It should be noted that in cases where the gradient has high values, the value for diffusion function $D(\nabla I) \rightarrow 0$ and the smoothing process is halted.

5. Spatial K-Means Clustering Algorithm (S-KM)

One of the main problems associated with standard clustering algorithms is the lack of using any spatial continuity with respect to the local texture and color information in the space partitioning process. Thus the application of these algorithms is restricted to input images that are defined by homogenous color regions. Our aim is to develop a space-partitioning algorithm that is able to return meaningful results even when applied to complex natural scenes that exhibit large variation in color and texture. In order to produce accurate non-fragmented segmented results, we need to sample the homogeneity of a color-texture descriptor in a given region. Nonetheless the robust evaluation of the texture is a very difficult task, since the texture is not constant within the image and this would require complex models to describe it at micro and macro level. As we are interested in evaluating the complexity of the image locally, we reformulate the problem since we do not need precise models for texture and rather the evaluation of the local image complexity in a data with a reduced number of color components. Using these assumptions, a large number of algorithms have been developed to address the problem of robust segmentation of complex images and the most representative are the mean shift [2], adaptive clustering algorithm [3] and the extension of the diffusion based algorithms [10]. Our algorithm is novel, since it attempts to minimize the errors in the assignment of the data points into clusters by evaluating the local texture complexity using two measures that constrain the region intensity (color smoothness) and the spatial continuity (texture complexity). In this regard, the clustering algorithms are perfectly suited to perform this task and the main difficulty resides in the selection of optimal features that are able to produce clusters with spatial homogeneity. In addition, these measures have to be sufficiently generic, to be able to accommodate the local variation in texture scale and the local variation in color. To address these problems we propose to use two types of descriptors along with color information. To sample the local color homogeneity, we evaluate the distribution of the data in the image resulting after the application of the diffusion filtering (see equation 4). The local texture

complexity is sampled by the gradient data that is calculated using the Laplace operator that has been chosen for its low computational cost and its omni-directional properties (this assures immunity to texture rotation). Thus, the process of space partitioning is modified to accommodate these two distributions that are calculated as follows:

$$H_{(n,m)} = \bigcup_{i \in C} h(i), \quad h(i) = \sum_{y=-n}^n \sum_{x=-m}^m \delta(f(x,y), i),$$

where C is the color space, $\delta(i, j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$ (6)

Using the equation 6, we calculate the distributions from diffused and gradient images in the local neighborhood (n, m) for each data point. The global objective function depicted in equation 1 is modified to accommodate the diffusion-gradient distributions as follows:

$$J = \sum_{j=1}^K \sum_{i=1}^n \left[\|x_i^{(j)} - c_j\| + KS(H_{Diff_i}^{(j)}, H_{Diff}^{(j)}) + KS(H_{Grad_i}^{(j)}, H_{Grad}^{(j)}) \right] \quad (7)$$

where $H_{Diff_i}^{(j)}$ is the local color smoothness distribution (calculated from diffused image) for the data point i , $H_{Diff}^{(j)}$ is the color smoothness distribution for cluster j , $H_{Grad_i}^{(j)}$ is the local texture complexity (calculated from the gradient data) for the data point i , $H_{Grad}^{(j)}$ is the texture complexity for cluster j , and KS is the Kolmogorov-Smirnov metric that is calculated using the following expression:

$$KS(H_a, H_b) = \sum_{i \in C} \left| \frac{h_a(i)}{n_a} - \frac{h_b(i)}{n_b} \right| \quad (8)$$

where the n_a and n_b are the number of data points in the distribution H_a and H_b respectively. The KS similarity measure is bounded in the interval $[0,2]$ and we have readjusted the metric $\|x_i^{(j)} - c_j\|$ to be also bounded (normalization with respect to the maximum value of the respective color space, i.e. 255 for RGB). The $H_{Diff}^{(j)}$ and $H_{Grad}^{(j)}$ distributions are recalculated after each iteration. It is important to note that during the space partitioning process (minimization of the objective function illustrated in equation 7) the number of clusters are reduced, since some clusters from the initial set will disappear as the clusters become more compact after each iteration. The developed algorithm is convergent and to improve the computational overhead we have applied the K-Means algorithm in the standard form for the first 5 iterations and then the spatial and color continuity constraints $H_{Diff}^{(j)}$ and $H_{Grad}^{(j)}$ are evaluated. This approach also improve the stability of the algorithm, since the $H_{Diff}^{(j)}$ and $H_{Grad}^{(j)}$ distributions can be calculated only after the algorithm executes at least one iteration.

6. Experiments and Results

In this section we examine the performance of the developed algorithm on a large number of images. The first test is performed on a synthetic color image that is corrupted with noise (standard deviation 30 grayscale values – see Figure 1). To assess the efficiency of our method, we compare the results against the segmentation results returned by the mean shift algorithm, which is widely accepted as the standard color segmentation framework. Additional results are illustrated in Figures 2 and 3, where the algorithm has been applied to a complex natural image and a natural image defined by a low signal to noise ratio. It can be observed that our algorithm produces better visual results than the mean shift algorithm. The parameters for our method are set as follows: initial number of clusters = 10 and diffusion parameter = 30 (all images). The parameters for Comanicu-Meer algorithm (Edison implementation) are set as follows: spatial filter=8, color=50 and minimum region size=200 when applied to the noisy image illustrated in Figure 1(a), spatial filter=7, color=6.5 and minimum region size=20 when the algorithm was applied to the image depicted in Figure 2(a) and spatial filter=7, color=15 and minimum region size=20 for the image depicted in Figure 3(a). The parameters for mean-shift algorithm are selected to generate the best results.

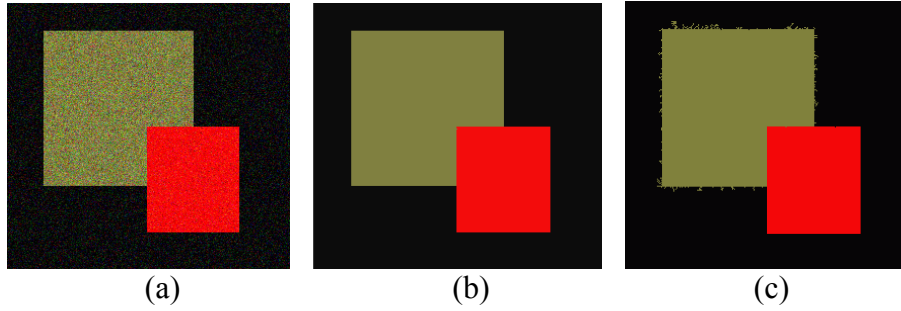


Figure 1. Segmentation results on a test image. (a) Test image corrupted with Gaussian noise (standard deviation 30 grayscales). (b) Segmented result – our algorithm (final number of regions=3). (c) Segmented result –mean shift algorithm (final number of regions=3).

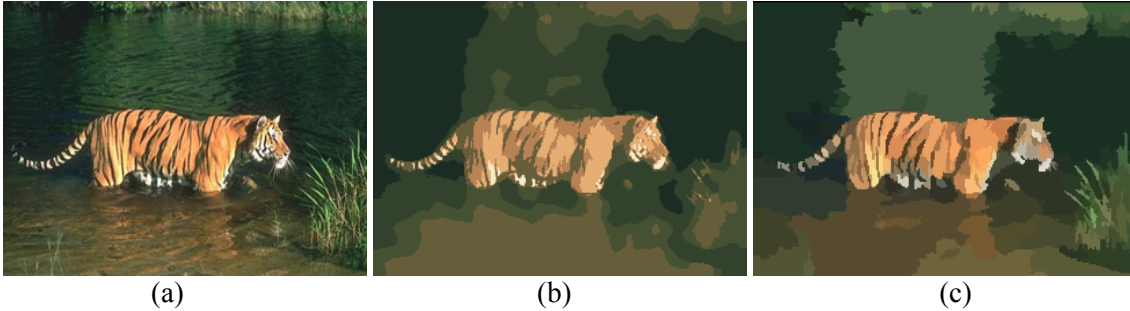


Figure 2. Segmentation results. (a) Input image. (b) Segmented result- our algorithm (final no of regions=7). (c) Segmented result – mean shift algorithm (final no of regions=223).

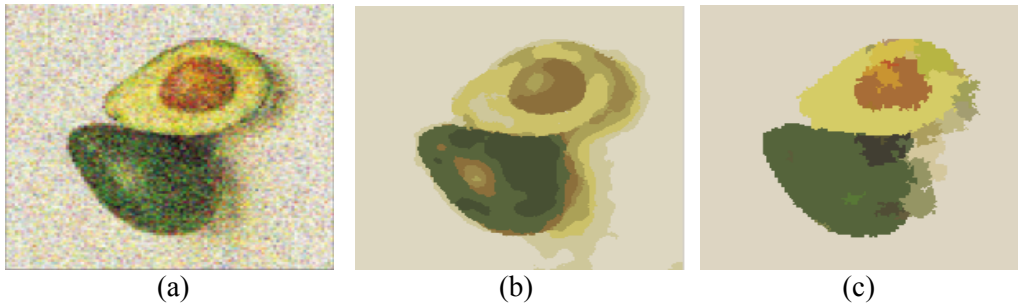


Figure 3. Segmentation results for a low-resolution natural image. (a) Input image. (b) Segmented result – our algorithm (final number of regions =6). (c) Segmented result – mean shift algorithm (final number of regions=27).

Another aim is to evaluate the behavior of our algorithm with respect to different color spaces. As expected, the RGB color space is non-linear and does not provide optimal color separation and our experiments indicate that better segmentation is achieved when the color images are converted to YIQ and HSI color spaces. Figures 4 and 5 depict the segmentation results when our algorithm has been applied to two test images. Our experiments indicate that best segmentation is obtained when the images are converted to YIQ color space. To fully evaluate the performance of our algorithm we evaluated the segmentation error for a natural image and for its version corrupted with additional noise (see Figure 6) when compared to the ground truth (manual segmentation), while the parameters of the algorithm (for this implementation the diffusion and gradient distributions are calculated within a square window, $n=m$) are varied. From the graphs illustrated in Figure 7, it can be noted the good stability of the developed algorithm when the parameters are varied. From these graphs it can also be observed that the segmentation error tends to be reduced with the increase of the diffusion parameter and this is generated by the fact that the smoothing is more pronounced and this translates into a more discriminative power for diffusion distribution in equation 7. The selection of the optimal window size is a difficult problem since a large window increases the discriminative power of the H_{Diff} and H_{Grad} distributions, but on the other hand increases the errors around the cluster borders. The optimal trade-off is achieved when the window size is set in the range $[7 \times 7, 11 \times 11]$.

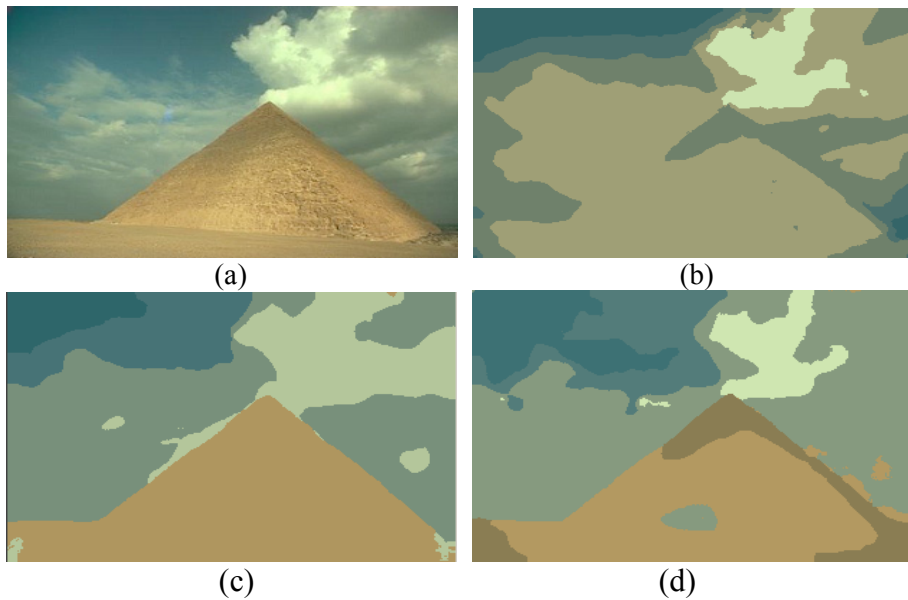


Figure 4. Segmentation results. (a) Test image. (b) Segmented result – RGB color space. (c) Segmented result – YIQ color space. (d) Segmented result – HSI color space.



Figure 5. Segmentation results. (a) Test image. (b) Segmented result – RGB color space. (c) Segmented result – YIQ color space. (d) Segmented result – HSI color space.

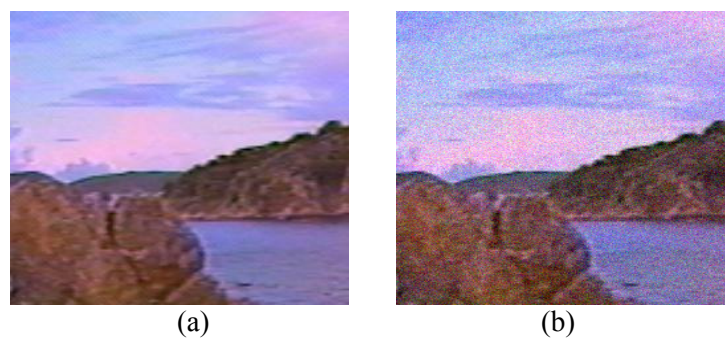


Figure 6. Test images used in the evaluation of the developed algorithm. (a) Test image with low resolution. (b) Image corrupted with additional noise (standard deviation 30 grayscales).

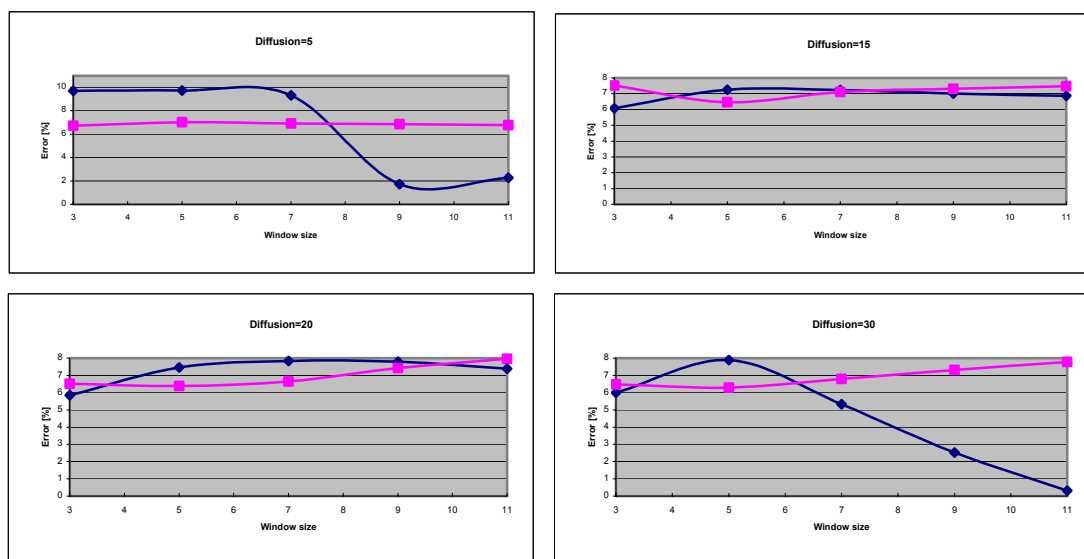


Figure 7. Performance of the developed algorithm when the diffusion and window size parameters are varied (blue dark noiseless image-Figure 6(a), purple noisy image-Figure 6(b)).

7. Conclusions

The aim of this work is the development of a new color segmentation algorithm, which is a generalization of the K-Means clustering algorithm. In its standard form the application of the K-Means algorithm to the segmentation of natural images is hindered by the fact that no constraints with respect to texture complexity or color continuity are employed during the space partitioning process. By reformulating the objective of the clustering process, we have developed a spatial constrained clustering algorithm that is found to be a powerful technique to identify continuous clusters in the color images that are strongly related with the scene objects. The developed algorithm has been tested and evaluated on a large number of natural images. The experimental data indicates that our algorithm is generic and it is robust to changes in local texture and produces accurate results even when applied to images characterized by a low signal to noise ratio.

Acknowledgments

We would like to express our gratitude to Science Foundation Ireland for supporting this research.

References

- [1] Deng Y. and Manjunath B.S. (2001). Unsupervised segmentation of color-texture regions in images and video, *IEEE Trans. Pattern Analysis Machine Intell*, 23(8): 800-810.
- [2] Comaniciu D. and Meer P. (2002). Mean shift: A robust approach toward feature space analysis”, *IEEE Trans. Pattern Analysis Machine Intell.*, 24(5): 603-619.
- [3] Pappas T. (1992). An adaptive clustering algorithm for image segmentation, *IEEE Trans. Signal Process*, 40:901-914.
- [4] Drimborean A. and Whelan P.F. (2001) Experiments in color texture analysis, *Pattern Recognition Letters*, 22: 1161-1167.
- [5] Zöllner H.T. and Buhmann J.M. (2002). Parametric distributional clustering for image segmentation, *7th European Conference on Computer Vision*, Denmark, pp. 577-591.
- [6] Mukherjee J. (2002). MRF clustering for segmentation of color images, *Pattern Recognition Letters*, 23(8): 917-929.
- [7] Khan S. and Ahmad A., (2004). Cluster center initialization algorithm for K-Means clustering, *Pattern Recognition Letters*, 25(11): 1293-1302.
- [8] Jain A.K. and Dubes R.C. (1988). *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ.
- [9] Frigui H. and Krishnapuram R. (1997). Clustering by competitive agglomeration, *Pattern Recognition*, 30(7): 1109-1119.
- [10] Perona P. and Malik J. (1990). Scale-space and edge detection using anisotropic diffusion, *IEEE Trans. Pattern Analysis Machine Intell*, 12(7): 629-639.